IN THE CLAIMS

1. (currently amended) A method of producing a simulation model of an electronic design, ~~which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design,~~ the method comprising:

receiving a non-obfuscated version of the electronic design suitable for direct compilation **into** ~~to~~ a practical hardware implementation of the electronic design; ~~and~~

adding obfuscation circuitry **to said electronic design** to produce an obfuscated version of the electronic design**, wherein said obfuscation circuitry prevents practical implementation of the electronic design on a target hardware device; and** ~~from which the simulation model can be created,~~

creating a simulation model using said obfuscated version of said electronic design, said simulation model being suitable for producing accurate hardware simulation results in a simulator but not being suitable to be directly compiled to produce a practical hardware implementation of the electronic design.

~~wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.~~

2. (currently amended) A method as recited in claim 1, wherein the non-obfuscated version of the electronic design is provided in **an** ~~a~~ HDL source format **and said creating a simulation model includes**

using said obfuscated version of said electronic design as said simulation model.

3. (currently amended) A method as recited in claim 1, wherein the electronic design is a reusable **functional logic block** ~~IP core~~.

4. (currently amended) A method as recited in claim 1, wherein adding obfuscation circuitry **includes** ~~comprises~~:

identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

choosing a type of obfuscation circuitry for insertion; and

inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

5. (currently amended) A method as recited in claim 4, wherein identifying a region for introduction of obfuscation circuitry **includes** ~~comprises~~ identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

6. (currently amended) A method as recited in claim 5, wherein the type of logic that is not removed by a synthesizer **includes** ~~comprises~~ one or more flip-flops.

7. (currently amended) A method as recited in claim 1, further **comprising:** ~~comprising~~

optimizing the obfuscated version of the electronic design **by merging** ~~to merge~~ the obfuscation circuitry with **non-obfuscated** functional circuitry **of said obfuscated version**.

8. (currently amended) A method as recited in claim 1, wherein the obfuscation circuitry ~~comprises circuitry that~~ increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

9. (original) A method as recited in claim 1, wherein adding obfuscation circuitry comprises:

at a first location, adding circuitry for scrambling an input signal by spreading out the input signal in time; and

at a second location, adding circuitry for de-scrambling an output signal resulting from the circuitry for scrambling.

10. (original) A method as recited in claim 1, wherein adding obfuscation circuitry comprises:

at a first location, adding circuitry for entangling multiple input signals to thereby spread out the input signals; and

at a second location, adding circuitry for detangling an output signal resulting from the circuitry for entangling.

11. (currently amended) A method as recited in claim 1, wherein the obfuscation circuitry **includes an** ~~comprises a~~ XOR tree.

12. (currently amended) A method as recited in claim 1, wherein adding obfuscation circuitry is performed automatically **without user intervention**.

13. (currently amended) An apparatus for producing a simulation model of an electronic design, ~~which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design,~~ the apparatus comprising:

one or more processors;

memory; ~~and~~

a design entry tool that allows a developer to input a non-obfuscated version of said electronic design;

an obfuscation module for adding obfuscation circuitry to **said** ~~a~~ non-obfuscated version of the electronic design to produce an obfuscated version of the electronic design from which the simulation model can be created**, wherein said obfuscation circuitry prevents practical implementation of the electronic design on a target hardware device, said obfuscation module creating said simulation model, said simulation model being suitable for producing accurate hardware simulation results in a simulator but not being suitable to be directly compiled to produce a practical hardware implementation of the electronic design.**

~~wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.~~

14. (currently amended) An apparatus as recited in claim 13, wherein the non-obfuscated version of the electronic design is in **an** ~~a~~ HDL source format**, said simulation model being said obfuscated version of said electronic design**.

15. (currently amended) An apparatus as recited in claim 13, wherein the electronic design is a reusable **functional logic block** ~~IP core~~.

16. (original) An apparatus as recited in claim 13, wherein the obfuscation module comprises:

a scanning module for identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

a selection module for choosing a type of obfuscation circuitry for insertion; and

an insertion module for inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

17. (currently amended) An apparatus as recited in claim 16, wherein the scanning module for identifying a region for introduction of obfuscation circuitry **includes** ~~comprises~~ identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

18. (currently amended) An apparatus as recited in claim 17, wherein the type of logic that is not removed by a synthesizer **includes** ~~comprises~~ one or more flip-flops.

19. (currently amended) An apparatus as recited in claim 13, further comprising:

an optimizer for optimizing the obfuscated version of the electronic design **by merging** ~~to merge~~ the obfuscation circuitry with **non-obfuscated** functional circuitry **of said obfuscated version**.

20. (currently amended) An apparatus as recited in claim 13, wherein the obfuscation circuitry ~~comprises circuitry that~~ increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

21. (currently amended) An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises ~~of~~:

at a first location, a scrambler having circuitry for scrambling an input signal by spreading out the input signal in time; and

at a second location, a descrambler having circuitry for de-scrambling an output signal resulting from the circuitry for scrambling.

22. (currently amended) An apparatus as recited in claim 13, wherein the obfuscation circuitry comprises ~~of~~:

at a first location, an entangler having circuitry for entangling multiple input signals to thereby spread out the input signals; and

at a second location, a detangler having circuitry for detangling an output signal resulting from the circuitry for entangling.

23. (currently amended) An apparatus as recited in claim 13, wherein the obfuscation circuitry **includes an** ~~comprises a~~ XOR tree.

24. (currently amended) An apparatus as recited in claim 16, wherein the scanning module, the selection module, and the insertion module are configured to operate automatically **without user intervention**.

25. (currently amended) A computer program product comprising a machine readable medium on which is provided program instructions for producing a simulation model of an electronic design, ~~which simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the electronic design,~~ the program instructions comprising:

instructions for receiving a non-obfuscated version of the electronic design suitable for direct compilation **into** ~~to~~ a practical hardware implementation of the electronic design; ~~and~~

instructions for adding obfuscation circuitry **to said electronic design** to produce an obfuscated version of the electronic design**, wherein said obfuscation circuitry prevents practical implementation of the electronic design on a target hardware device; and** ~~from which the simulation model can be created,~~

instructions for creating a simulation model using said obfuscated version of said electronic design, said simulation model being suitable for producing accurate hardware simulation results in a simulator but not being suitable to be directly compiled to produce a practical hardware implementation of the electronic design.

~~wherein the obfuscation circuitry does not substantially impact the accuracy of the simulation result, but prevents practical implementation of the electronic design on a hardware device.~~

26. (currently amended) A computer program product as recited in claim 25, wherein the non-obfuscated version of the electronic design is provided in **an** ~~a~~ HDL source format **and said creating a simulation model includes**

using said obfuscated version of said electronic design as said simulation model.

27. (currently amended) A computer program product as recited in claim 25, wherein the electronic design is a reusable **functional logic block** ~~IP core~~.

28. (currently amended) A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry comprises:

instructions for identifying a region for introduction of obfuscation circuitry in the non-obfuscated version of the electronic design;

instructions for choosing a type of obfuscation circuitry for insertion; and

instructions for inserting the chosen type of obfuscation circuitry into the identified region, thereby creating an obfuscated region.

29. (currently amended) A computer program product as recited in claim 28, wherein the instructions for identifying a region for introduction of obfuscation circuitry comprises identifying in the non-obfuscated version of the electronic design logic of a type that is not removed by a synthesizer.

30. (currently amended) A computer program product as recited in claim 29, wherein the type of logic that is not removed by a synthesizer **includes** ~~comprises~~ one or more flip-flops.

31. (currently amended) A computer program product as recited in claim 28, further comprising:

instructions for optimizing the obfuscated version of the electronic design **by merging** ~~to merge~~ the obfuscation circuitry with **non-obfuscated** functional circuitry **of said obfuscated version**.

32. (currently amended) A computer program product as recited in claim 25, wherein the obfuscation circuitry ~~comprises circuitry that~~ increases the size of the electronic design without changing its function and/or slows the speed of the electronic design without changing its function.

33. (currently amended) A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry **comprise** ~~comprises~~:

instructions for adding circuitry at a first location to scramble an input signal by spreading out the input signal in time; and

instructions for adding circuitry at a second location to de-scrambling an output signal resulting from the circuitry to scramble.

34. (currently amended) A computer program product as recited in claim 25, wherein the instructions for adding obfuscation circuitry **comprise** ~~comprises~~:

instructions for adding circuitry at a first location to entangle multiple input signals to thereby spread out the input signals; and

instructions for adding circuitry at a second location to detangle an output signal resulting from the circuitry to entangle.

35. (currently amended) A computer program product as recited in claim 25, wherein the obfuscation circuitry **includes an** ~~comprises~~ a XOR tree.

36. (currently amended)A computer program product as recited in claim 28, wherein the operations of identifying, choosing, and inserting can be done automatically **without user intervention**.

37. (currently amended) A method of producing a simulation model of an **intellectual property** ~~IP~~ core, wherein the simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the IP core, the method comprising:

(a) receiving a non-obfuscated version of the IP core in a native HDL format or in a partially compiled HDL format;

(b) identifying a region of the non-obfuscated IP core where one or more flip-flops are located;

(c) inserting entangler circuitry upstream from the region and inserting complementary detangler circuitry downstream from the region;

(d) inserting scrambler circuitry upstream from the region and inserting complementary descrambler circuitry downstream from the region; ~~and~~

(e) optimizing the IP core after the insertions of (c) and (d)**; and**

(f) producing a simulation model using said optimized IP core that includes said inserted entangler and inserted scrambler circuitry.

38. (currently amended) A method of producing a simulation model of an **intellectual property** ~~IP~~ core, wherein the simulation model produces a hardware simulation result but cannot be directly compiled to produce a practical hardware implementation of the IP core, the method comprising:

(a) receiving a non-obfuscated version of the IP core in a native HDL format or in a partially compiled HDL format;

(b) identifying a region of the non-obfuscated IP core where one or more flip-flops are located;

(c) inserting obfuscation circuitry into the region;

(d) adding additional flip-flops and/or modifying the flip-flops; and

(e) optimizing the IP core after (c) and (d) have been performed**; and**

(f) producing a simulation model using said optimized IP core that includes said inserted obfuscation circuitry.

39. (currently amended) ~~An IP core~~ **A computer program product comprising a machine readable medium on which is provided program instructions for implementing an intellectual property (IP) core, said program instructions** comprising:

a programming version of the IP core for insertion in an electronic design developed using a specified **electronic design automation (EDA)** ~~EDA~~ platform; and

a simulation model of the IP core for simulating operation of the IP core in the electronic design, wherein the simulation model **includes** ~~comprises~~ obfuscation circuitry, absent in the programming version, which allows **an accurate** ~~a~~ hardware simulation result of the IP core but prevents direct compilation **of the simulation model** to produce a practical hardware implementation of the IP core.

Please add the following new claims:

40. (New) A method as recited in claim 8, wherein the obfuscation circuitry substantially increases the area of the electronic design or reduces the speed of a critical path of the electronic design.

41. (New) An apparatus as recited in claim 20, wherein the obfuscation circuitry substantially increases the area of the electronic design or reduces the speed of a critical path of the electronic design.

42. (New) A computer program product as recited in claim 32, wherein the obfuscation circuitry substantially increases the area of the electronic design or reduces the speed of a critical path of the electronic design.

43. (New) A method as recited in claim 37, wherein said entangler and scrambler circuitry substantially increases the area of said functional logic block and reduces the speed of a critical path of said functional logic block.

44. (New) A method as recited in claim 38, wherein said obfuscation circuitry and said additional flip-flops substantially increase the area of the electronic design.

45. (New) An intellectual property core as recited in claim 39, wherein said obfuscation circuitry substantially increases the area of said intellectual property core or reduces the speed of a critical path of said intellectual property core.

46. (New) A method as recited in claim 1 wherein said simulation model is cycle accurate and bit accurate.

47. (New) An apparatus as recited in claim 13 wherein said simulation model is cycle accurate and bit accurate.

48. (New) A computer program product as recited in claim 25 wherein said simulation model is cycle accurate and bit accurate.

49. (New) A method as recited in claim 37 further comprising:

producing a simulation model using said optimized intellectual property core, wherein said simulation model is cycle accurate and bit accurate.

50. (New) A method as recited in claim 38 further comprising:

producing a simulation model using said optimized intellectual property core, wherein said simulation model is cycle accurate and bit accurate.

51. (New) An intellectual property core as recited in claim 39 wherein said simulation model is cycle accurate and bit accurate.

52. (New) A method as recited in claim 3 wherein said functional logic block is an intellectual property core.

53. (New) An apparatus as recited in claim 15 wherein said functional logic block is an intellectual property core.

54. (New) A computer program product as recited in claim 27 wherein said functional logic block is an intellectual property core.

55. (New) A method as recited in claim 1, wherein the non-obfuscated version of the electronic design is provided in a partially compiled format, and wherein said creating a simulation model includes

using a translation utility to convert said obfuscated version of said electronic design into said simulation model having a standard format usable by a variety of simulators.

56. (New) An apparatus as recited in claim 13, wherein the non-obfuscated version of the electronic design is provided in a partially compiled format, said apparatus further comprising:

a model writer module that converts said obfuscated version of said electronic design into said simulation model having a standard format usable by a variety of simulators.

57. (New) A computer program product as recited in claim 25, wherein the non-obfuscated version of the electronic design is provided in a partially compiled format, and wherein said creating a simulation model includes

using a translation utility to convert said obfuscated version of said electronic design into said simulation model having a standard format usable by a variety of simulators.